

2026 年信息素养大赛 C++小学组模拟卷二

一、选择题（每题 4 分，共 60 分）

1. 以下程序使用冒泡排序对一组表示汉字笔画数的数组进行排序，程序运行后数组 arr 的内容是什么？

```
#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main() {
    int arr[] = {8, 3, 12, 5, 7}; // 分别代表“永”8画、“山”3画、“鼎”12画、“王”5画、“我”7画

    int n = sizeof(arr)/sizeof(arr[0]);
```

```

bubbleSort(arr, n);

// 输出排序结果

for (int i = 0; i < n; i++) {
    cout << arr[i] << " ";
}

return 0;
}

```

- A. 3 5 7 8 12
- B. 12 8 7 5 3
- C. 3 8 5 12 7
- D. 8 3 12 5 7

2. 在存储中国历史朝代建立年份的有序数组中，使用二分查找算法查找唐朝建立年份（618 年）。给定数组 `years[] = {221, 420, 581, 618, 960, 1271, 1368, 1644}`，查找过程中依次比较的元素是？

```

int binarySearch(int arr[], int n, int target) {
    int left = 0, right = n-1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        cout << arr[mid] << " "; // 输出比较的元素
        if (arr[mid] == target) return mid;
        else if (arr[mid] < target) left = mid + 1;
        else right = mid - 1;
    }
}

```

```
    }  
  
    return -1;  
}
```

- A. 581 960 618
- B. 581 618
- C. 960 581 618
- D. 618

3. 以下程序处理成语“画龙点睛”，输出结果是什么？

```
#include <iostream>  
  
#include <string>  
  
using namespace std;  
  
int main() {  
  
    string idiom = "画龙点睛";  
  
    string result = "";  
  
    for (int i = 0; i < idiom.length(); i++) {  
        if (i % 2 == 0) {  
            result += idiom[i];  
        }  
    }  
  
    result += " -> ";  
  
    for (int i = idiom.length() - 1; i >= 0; i--) {
```

```
        result += idiom[i];  
    }  
  
    cout << result << endl;  
  
    return 0;  
}
```

- A. 画点 -> 睛点龙画
- B. 画睛 -> 点睛龙画
- C. 画点 -> 点睛龙画
- D. 画睛 -> 睛点龙画

4. 使用选择排序对历史人物去世年龄进行排序，数组 `ages[] = {72, 66, 56, 65, 70}`（孔子 72 岁、李白 66 岁、杜甫 56 岁、苏轼 65 岁、陆游 70 岁）。第一轮排序后数组状态是？

```
void selectionSort(int arr[], int n) {  
    for (int i = 0; i < n-1; i++) {  
        int minIdx = i;  
        for (int j = i+1; j < n; j++) {  
            if (arr[j] < arr[minIdx]) {  
                minIdx = j;  
            }  
        }  
  
        int temp = arr[minIdx];  
        arr[minIdx] = arr[i];  
        arr[i] = temp;  
    }  
}
```

```
arr[i] = temp;

// 第一轮结束后输出

if (i == 0) {

    for (int k = 0; k < n; k++) cout << arr[k] << " ";

    cout << endl;

}

}
```

- A. 56 66 72 65 70
- B. 56 72 66 65 70
- C. 72 56 66 65 70
- D. 66 72 56 65 70

5. 以下表达式计算农历闰月情况，设 year=2023，leapMonth=2，表达式 $(year \% 4 == 0 \ \&\& \ year \% 100 != 0) \ || \ (year \% 400 == 0)$ ？

leapMonth + 1 : leapMonth 的结果是？

- A. 2
- B. 3
- C. 4
- D. 5

6. 以下函数统计古诗中指定字符出现的次数，调用 `countChar("床前明月光，疑是地上霜。", '月')` 的返回值是？

```
int countChar(string poem, char ch) {  
    int count = 0;  
    for (int i = 0; i < poem.length(); i++) {  
        if (poem[i] == ch) {  
            count++;  
        }  
    }  
    return count;  
}
```

- A. 1
- B. 2
- C. 3
- D. 4

7. 归并排序在合并两个已排序的历史事件年份数组时，`left[] = {1368, 1644}`，`right[] = {1912, 1949}`，合并后的数组是？

```
vector<int> merge(vector<int> left, vector<int> right) {  
    vector<int> result;  
    int i = 0, j = 0;  
    while (i < left.size() && j < right.size()) {  
        if (left[i] <= right[j]) {
```

```

        result.push_back(left[i++]);
    } else {
        result.push_back(right[j++]);
    }
}

while (i < left.size()) result.push_back(left[i++]);
while (j < right.size()) result.push_back(right[j++]);
return result;
}

```

- A. 1368 1644 1912 1949
- B. 1368 1912 1644 1949
- C. 1912 1368 1644 1949
- D. 1368 1644 1949 1912

8. 字符串“苗族-壮族-回族-满族”经过以下程序处理后，输出是什么？

```

string ethnicGroups = "苗族-壮族-回族-满族";
string delimiter = "-";
size_t pos = 0;
string token;
while ((pos = ethnicGroups.find(delimiter)) != string::npos)
{
    token = ethnicGroups.substr(0, pos);
}

```

```
    cout << token.length() << " ";

    ethnicGroups.erase(0, pos + delimiter.length());
}

cout << ethnicGroups.length() << endl;
```

- A. 2 2 2 2
- B. 2 4 6 8
- C. 2 2 2
- D. 2 2 2 2 2

9. 传统建筑结构体

以下结构体表示传统建筑，程序输出是什么？

```
#include <iostream>

#include <string>

using namespace std;

struct AncientBuilding {

    string name;

    int year;

    string dynasty;

};

int main() {

    AncientBuilding building = {"岳阳楼", 1044, "北宋"};

    AncientBuilding* ptr = &building;

    cout << ptr->name << "建于" << ptr->dynasty << endl;
```

```
    return 0;
}
```

- A. 岳阳楼建于北宋
- B. building 建于北宋
- C. 1044 建于北宋
- D. 编译错误

10. 在已排序的战役年份数组 `battles[] = {208, 383, 618, 755, 1127, 1368}` 中查找安史之乱年份 755，二分查找需要比较多少次？

- A. 1 次
- B. 2 次
- C. 3 次
- D. 4 次

11. 传统文化多重循环

以下嵌套循环输出什么图案？

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= i; j++) {
        cout << "★";
    }
    cout << endl;
}
```

- A.

★★★★

★★

★

B.

★

★★

★★★★

C.

★★★★

★★★★

★★★★

D. ★

★★

★★★★

12. 在优化版的冒泡排序中，如果某一轮没有发生交换，则说明数组已有序。对数组[1, 2, 3, 4, 5]进行冒泡排序，需要多少轮循环？

```
void optimizedBubbleSort(int arr[], int n) {
```

```
    bool swapped;
```

```
    for (int i = 0; i < n-1; i++) {
```

```
        swapped = false;
```

```
        for (int j = 0; j < n-i-1; j++) {
```

```
            if (arr[j] > arr[j+1]) {
```

```
        swap(arr[j], arr[j+1]);

        swapped = true;
    }

    if (!swapped) break;
}
}
```

- A. 1 轮
- B. 2 轮
- C. 4 轮
- D. 5 轮

13. 以下两个函数构成重载关系，调用 `calculate(5, 3)` 会执行哪个函数？

```
int calculate(int a, int b) {
    return a + b; // 加法，如“五行相生”
}

double calculate(double a, double b) {
    return a * b; // 乘法，如“天地相乘”
}
```

- A. 第一个函数，返回 8
- B. 第二个函数，返回 15.0

- C. 编译错误
- D. 运行时错误

14. 以下程序输出结果是什么？

```
string dynasty1 = "唐";  
string dynasty2 = "宋";  
string dynasty3 = "元";  
string result = dynasty1 + "->" + dynasty2 + "->" + dynasty  
3;  
cout << result.substr(2, 3) << endl;
```

- A. 唐->宋
- B. ->宋->
- C. 宋->元
- D. ->宋

15. 以下程序按历史人物出生年份排序，输出第二个人物的名字：

```
Person persons[] = {{ "孔子", -551}, {"李白", 701}, {"苏轼",  
1037}}; 按年份升序排序后  
struct Person {  
    string name;  
    int birthYear;  
};
```

// 使用选择排序按 *birthYear* 排序

```
void sortPersons(Person arr[], int n) {  
    for (int i = 0; i < n-1; i++) {  
        int minIdx = i;  
        for (int j = i+1; j < n; j++) {  
            if (arr[j].birthYear < arr[minIdx].birthYear) {  
                minIdx = j;  
            }  
        }  
        swap(arr[i], arr[minIdx]);  
    }  
}
```

// 排序后输出 *arr[1].name*

- A. 孔子
- B. 李白
- C. 苏轼
- D. 无法确定

二、判断题（每题 2 分，共 10 分）

1. 冒泡排序中，每一轮都将最大的元素移动到正确位置。

- A. 正确
- B. 错误

2. 二分查找算法要求数组必须完全无序才能正常工作。

A. 正确

B. 错误

3. 在 C++ 中，字符串“端午节”的长度是 6 个字符，因为每个汉字占 2 个字节。

A. 正确

B. 错误

4. 归并排序的时间复杂度在最坏情况下是 $O(n^2)$ 。

A. 正确

B. 错误

5. 结构体的大小等于所有成员变量大小之和。（）

A. 正确

B. 错误